

Application No. 09/872,581
RCE to OA of 06/09/2006

Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of the claims:

1. (currently amended) A method for invoking multiple parallel instances of a same node comprising the steps of:
 - a) defining a multinode as a node that allows for activation of multiple parallel instances of a same work node in a workflow;
 - b) ~~during~~ at run time determining a number of work nodes to be activated in the multinode ~~equal to~~ ~~based on~~ a number of elements in a vector;
 - c) ~~activating~~ ~~executing~~ the number of work nodes in the multinode;
 - d) determining when the execution of the multinode is completed based on a termination rule regardless of a status of work nodes in the work flow;
 - e) when the execution of the multinode is complete, executing a successor node;and
 - f) when the execution of the multinode is not complete, processing continues at step c).
2. (previously presented) The method of claim 1
wherein a new instance of the same work node is started for the number of elements in the vector.
3. (canceled)
4. (previously presented) The method of claim 1
wherein determining when the execution of the multinode is completed based on said termination rule includes evaluating whether a multinode goal has been achieved, and when the multinode goal has been achieved, terminating the execution of the multinode.
5. (previously presented) The method of claim 1

Application No. 09/872,581
RCE to OA of 06/09/2006

wherein determining when the execution of the multinode is completed based on said termination rule includes determining whether all work nodes in the multinode have been completed, and when all work nodes in the multinode have been completed, terminating the execution of the multinode.

6. (previously presented) The method of claim 1

wherein terminating the execution of the multinode includes canceling other nodes and proceeding to a successor node.

7. (previously presented) The method of claim 1 further comprising the step of

allowing flow to continue to said successor node when all activated work nodes in the multinode have been completed.

8. (previously presented) The method of claim 1 wherein the step of

executing the multinode includes the step of providing each work node in the multinode with different input data for execution.

9. (previously presented) The method of claim 1 wherein the step of

executing the multinode includes the step of providing different attributes for each work node in the multinode.

10. (previously presented) The method of claim 9 wherein the attributes includes one of resource selection criteria, security, exception handling criteria, and deadlines for work node execution.

11. (previously presented) The method of claim 1 wherein the step of determining when the execution of the multinode is completed based on said termination rule further includes the step of

specifying multinode termination by a condition;

checking the condition when one of the work nodes in the multinode terminates;

and

Application No. 09/872,581
RCE to OA of 06/09/2006

when the condition is satisfied, said successor node is activated, and other work nodes in execution within the multinode are canceled.

12. (currently amended) A system for processing multinode definitions comprising:
a workflow engine for processing workflow definitions; and
a multinode handling facility coupled to the workflow engine for processing multinodes, determining a number of multiple instances of same work nodes in one of the multinodes to be activated equal to ~~based on~~ a number of elements in a vector; ~~activating~~ ~~executing~~ the number of work nodes in the one of the multinodes; determining when the execution of the one of the multinodes is completed based on a termination rule; and when the execution of the one of the multinodes is complete, executing a successor node.
13. (previously presented) The system of claim 12 wherein the multinode handling facility further comprises
a multinode determination unit for receiving a node definition and responsive thereto for determining whether the current node is a normal work node or one of the multinodes.
14. (previously presented) The system of claim 12 wherein the multinode handling facility further comprises
an activation facility for receiving an activation rule and based thereon for determining whether activation is by resource or by variable.
15. (previously presented) The system of claim 14 wherein the multinode handling facility further comprises
a resource-based activation facility coupled to the activation facility for processing activation by resources; and
a variable-based activation facility coupled to the activation facility for processing activation by variables.

Application No. 09/872,581
RCE to OA of 06/09/2006

16. (previously presented) The system of claim 15 wherein the resource-based activation facility further comprises
- a resource rule execution unit for executing a resource rule of the one of the multinodes; and
 - a new instance generation unit for starting new instances of the one of the multinodes for each new resource in the resource list.
17. (original) The system of claim 16 wherein the resource rule is specified in a service node tag of a multinode description.
18. (previously presented) The system of claim 15 wherein the variable-based activation facility further comprises
- a variable name reader for reading the variable name; and
 - a new instance generation unit for starting new instances of the one of the multinodes for each new element in the variable identified by the variable name.
19. (previously presented) The system of claim 18 wherein a variable type is one of a vector and a list.
20. (previously presented) The system of claim 12 wherein each node in the one of the multinodes is provided with different input data and different attributes for execution; wherein the attributes includes one of resource selection criteria, security, exception handling criteria, and deadlines for node execution.
21. (previously presented) A method for invoking multiple parallel instances of a same node, comprising:
- defining a multinode as a node that allows for activation of multiple parallel instances of a same work node in a workflow;
 - determining, based on an activation rule, whether the activation of the multiple parallel instances of the same work node is a resource-based activation or a variable-based activation; and

Application No. 09/872,581
RCE to OA of 06/09/2006

activating the multinode to execute plural instances of the same work node in the workflow.

22. (previously presented) The method of claim 21 further comprising, reading a variable name to determine a number of the multiple parallel instances of the same work node to activate when the activation is based on the variable-based activation.

23. (previously presented) The method of claim 21 further comprising, activating new instances of the multinode for each element in a vector when the activation is based on the variable-based activation.

24. (previously presented) The method of claim 21 wherein, the resource-based activation is based on a number of resources available for executing the workflow, and the variable-based activation is based on a vector.

25. (previously presented) A method for invoking multiple parallel instances of a same node, comprising:

defining a multinode as a node that allows for activation of multiple parallel instances of a same work node in a workflow;

reading an activation rule to determine if activation of the multinode is based on resource-based activation or variable-based activation; and

activating the multinode to execute a number of plural instances of the same work node, wherein (1) the number equals resources available for executing the workflow when the activation rule is the resource-based activation or (2) the number is read from a variable name that is a vector when the activation rule is the variable-based activation.

26. (previously presented) The method of claim 25, wherein a new instance of the same work node is started for each element in the vector.